

Block-Wise MAP Inference for Determinantal Point Processes with Application to Change-Point Detection

Jinye Zhang and Zhijian Ou

Speech Processing and Machine Intelligence Laboratory
Tsinghua University, Beijing, China, 100085

Abstract

Existing MAP inference algorithms for determinantal point processes (DPPs) need to calculate determinants or conduct eigenvalue decomposition generally at the scale of the full kernel, which presents a great challenge for real-world applications. In this paper, we introduce a class of DPPs, called BwDPPs, that are characterized by an almost block diagonal kernel matrix and thus can allow efficient block-wise MAP inference. Furthermore, BwDPPs are successfully applied to address the difficulty of selecting change-points in the problem of change-point detection (CPD), which results in a new BwDPP-based CPD method, named BwDppCpd. In BwDppCpd, a preliminary set of change-point candidates is first created based on existing well-studied metrics. Then, these change-point candidates are treated as DPP items, and DPP-based subset selection is conducted to give the final estimate of the change-points that favours both quality and diversity. The effectiveness of BwDppCpd is demonstrated through extensive experiments on five real-world datasets.

Introduction

The determinantal point processes (DPPs) are elegant probabilistic models for subset selection problems where both quality and diversity are considered. Formally, given a set of items $\mathcal{Y} = \{1, \dots, N\}$, a DPP defines a probability measure \mathcal{P} on $2^{\mathcal{Y}}$, the set of all subsets of \mathcal{Y} . For every subset $Y \subseteq \mathcal{Y}$ we have

$$\mathcal{P}_{\mathbf{L}}(Y) \propto \det(\mathbf{L}_Y), \quad (1)$$

where the L-ensemble kernel \mathbf{L} is an N by N positive semi-definite matrix. By writing $\mathbf{L} = \mathbf{B}^T \mathbf{B}$ as a Gram matrix, $\det(\mathbf{L}_Y)$ could be viewed as the squared volume spanned by the column vectors \mathbf{B}_i for $i \in Y$. By defining $\mathbf{B}_i = q_i \phi_i$, a popular decomposition of the kernel is given as

$$L_{ij} = q_i \phi_i^T \phi_j q_j, \quad (2)$$

where $q_i \in \mathbb{R}^+$ measures the quality (magnitude) of item i in \mathcal{Y} , and $\phi_i \in \mathbb{R}^k$, $\|\phi_i\| = 1$ can be viewed as the angle vector of diversity features so that $\phi_i^T \phi_j$ measures the similarity between items i and j . It can be shown that the probability of including i and j increases with the quality of i and j and diversity between i and j . As a result, a DPP assigns high probability to subsets that are both of good quality and diverse (Kulesza and Taskar 2012).

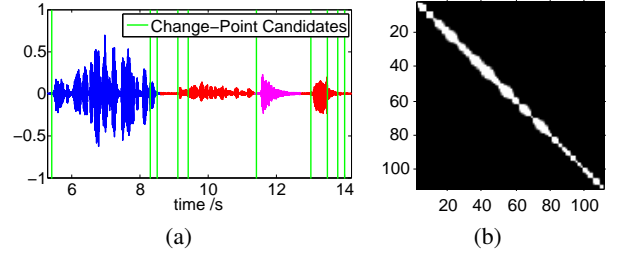


Figure 1: (a) A 10-sec part of a 2-min speech recording, shown with change-point candidates. Segments of different speakers or noises are plotted in different colors. (b) BwDPP kernel constructed for the whole 2-min recording, with the 112 change-point candidates as BwDPP items. The white denotes non-zero entries while the black indicates zero.

For DPPs, the *maximum a posteriori* (MAP) problem $\operatorname{argmax}_{Y \subseteq \mathcal{Y}} \det(\mathbf{L}_Y)$, aiming at finding the subset with highest probability, has attracted much attention due to its broad range for potential applications. Noting that this is an NP-hard problem (Ko, Lee, and Queyranne 1995), a number of approximate inference methods have been purposed, including the greedy methods for optimizing the submodular function $\log \det(\mathbf{L}_Y)$ (Buchbinder et al. 2012; Nemhauser, Wolsey, and Fisher 1978), optimization via continuous relaxation (Gillenwater, Kulesza, and Taskar 2012), and minimum Bayes risk decoding that minimizes the application-specific loss function (Kulesza and Taskar 2012).

These existing methods need to calculate determinants or conduct eigenvalue decomposition. Both computations are taken at the scale of the kernel size N and with the cost of around $\mathcal{O}(N^3)$ time that become intolerably high when N become large, e.g. thousands. Nevertheless, we find that for a class of DPPs where the kernel is almost block diagonal (Fig. 1 (b)), the MAP inference with the whole kernel could be replaced by a series of sub-inferences with its sub-kernels. Since the sizes of the sub-kernels become smaller, the overall computational cost can be significantly reduced. Such DPPs are often defined over a line where items are only similar to their neighbourhoods on the line and significantly different from those far away. Since the MAP inference for such DPPs is conducted in a block-wise manner, we refer to them as BwDPPs (block-wise DPPs) in the rest of the paper.

The above observation is mainly motivated by the prob-

lem of change-point detection (CPD) that aims at detecting abrupt changes in time-series data (Gustafsson and Gustafsson 2000). In CPD, the period of time between two consecutive change-points, often referred to as a segment or a state, is with homogeneous properties of interest (e.g. the same speaker in a speech (Chen and Gopalakrishnan 1998) or the same behaviour in human activity data (Liu et al. 2013)). After choosing a number of change-point candidates without much difficulty, we can treat these change-point candidates as DPP items, and select a subset from them to be our final estimate of the change-points. Each change-point candidate has its own quality of being a change-point. Moreover, the true locations of change-points along the timeline tend to be diverse, since states (e.g. speakers in Fig. 1 (a)) would not change rapidly. Therefore, it is preferred to conduct change-point selection that incorporates both quality and diversity. DPP-based subset selection clearly suits this purpose well. Meanwhile, the corresponding kernel will then become almost block diagonal (e.g. Fig. 1 (b)), as neighbouring items are less diversified, and items far apart more diversified. In this case, the DPP becomes BwDPP.

The problem of CPD have been actively studied for decades, where various CPD methods could be broadly classified into Bayesian or frequentist approach. In Bayesian approach, the CPD problem is reduced to estimating the posterior distribution of the change-point locations given the time-series data (Green 1995). Other posteriors to be estimated include the 0/1 indicator sequence (Lavielle and Lebarbier 2001), and the “run length” (Adams and MacKay 2007). Although many improvements were made, e.g. using advanced Monte Carlo method, the efficiency for estimating these posteriors is still a big challenge for real-world tasks.

In frequentist approach, the core idea is hypothesis testing and the general strategy is to first define a metric (test statistic) by considering the observations over past and present windows. As both windows move forward, change-points are selected when the metric value exceeds a threshold. Some widely-used metrics include the cumulative sum (Basseville, Nikiforov, and others 1993), the generalized likelihood-ratio (Gustafsson 1996), the Bayesian information criterion (BIC) (Chen and Gopalakrishnan 1998), the Kullback Leibler divergence (Delacourt and Wellekens 2000), and more recently, subspace-based metrics (Idé and Tsuda 2007; Kawahara, Yairi, and Machida 2007), kernel-based metrics (Desobry, Davy, and Doncarli 2005), and density-ratio (Kanamori, Suzuki, and Sugiyama 2010; Kawahara and Sugiyama 2012). While various metrics have been explored, how to choose thresholds and perform change-point selection, which is also a determining factor for detection performance, is relatively less studied. Heuristic-based rules or procedures are dominant and not well-performed, e.g. selecting local peaks above a threshold (Kawahara, Yairi, and Machida 2007), discarding the lower one if two peaks are close (Liu et al. 2013), or requiring the metric differences between change-points and their neighbouring valleys above a threshold (Delacourt and Wellekens 2000).

In this paper, we propose to apply DPP to address the difficulty of selecting change-points. Based on existing well-

studied metrics, we can create a preliminary set of change-point candidates without much difficulty. Then, we treat these change-point candidates as DPP items, and conduct DPP-based subset selection to obtain the final estimate of the change-points that favours both quality and diversity.

The contribution of this paper is two-fold. First, we introduce a class of DPP, called BwDPPs, that are characterized by an almost block diagonal kernel matrix and thus can allow efficient block-wise MAP inference. Second, BwDPPs are successfully applied to address the difficult problem of selecting change-points, which results in a new BwDPP-based CPD method, named BwDppCpd.

The rest of the paper is organized as follows. After describing brief preliminaries, we introduce BwDPPs and give our theoretical result on the BwDPP-MAP method. Next, we introduce BwDppCpd and present evaluation experiment results on a number of real-world datasets. Finally, we conclude the paper with a discussion on potential future directions.

Preliminaries

Throughout the paper, we are interested in MAP inference for BwDPPs, a particular class of DPP where the L-ensemble kernel \mathbf{L} is almost block diagonal¹, namely

$$\mathbf{L} \triangleq \begin{bmatrix} \mathbf{L}_1 & \mathbf{A}_1 & & \cdots & \mathbf{0} \\ \mathbf{A}_1^T & \mathbf{L}_2 & \mathbf{A}_2 & & \\ & \ddots & \ddots & \ddots & \vdots \\ & & \mathbf{A}_{m-2}^T & \mathbf{L}_{m-1} & \mathbf{A}_{m-1} \\ \mathbf{0} & \cdots & & \mathbf{A}_{m-1}^T & \mathbf{L}_m \end{bmatrix}, \quad (3)$$

where the diagonal sub-matrices $\mathbf{L}_i \in \mathbb{R}^{l_i \times l_i}$ are sub-kernels containing DPP items that are mutually similar, and the off-diagonal sub-matrices $\mathbf{A}_i \in \mathbb{R}^{l_i \times l_{i+1}}$ are sparse sub-matrices with non-zero entries only at the bottom left, representing the connections between adjacent sub-kernels. Fig. 2 (a) gives a good example of such matrices.

Let \mathcal{Y} be the set of all indices of \mathbf{L} and let $\mathcal{Y}_1, \dots, \mathcal{Y}_m$ be that of $\mathbf{L}_1, \dots, \mathbf{L}_m$ correspondingly. For any set of indices $C_i, C_j \subseteq \mathcal{Y}$, we use \mathbf{L}_{C_i} to denote the square sub-matrix indexed by C_i and \mathbf{L}_{C_i, C_j} the $|C_i| \times |C_j|$ sub-matrix with rows indexed by C_i and columns by C_j . Following general notations, by $\mathbf{L} = \text{diag}(\mathbf{L}_1, \dots, \mathbf{L}_m)$ we mean the block diagonal matrix \mathbf{L} consisting of sub-matrices $\mathbf{L}_1, \dots, \mathbf{L}_m$ and $\mathbf{L} \succeq 0$ means that \mathbf{L} is positive semi-definite.

MAP Inference for BwDPPs

Strictly Block Diagonal Kernel

We first consider the motivating case where the kernel is strictly block diagonal, i.e. all elements in the off-diagonal sub-matrices \mathbf{A}_i are zero. It can be easily seen that the following divide-and-conquer theorem holds.

Theorem 1 *For the DPP with a block diagonal kernel $\mathbf{L} = \text{diag}(\mathbf{L}_1, \dots, \mathbf{L}_m)$ over ground set $\mathcal{Y} = \bigcup_{i=1}^m \mathcal{Y}_i$ which is*

¹Such matrices could also be defined as a particular class of block tridiagonal matrices, where the off-diagonal sub-matrices \mathbf{A}_i only have a few non-zeros entries at the bottom left.

partitioned correspondingly, the MAP solution can be obtained as:

$$\hat{C} = \hat{C}_1 \cup \dots \cup \hat{C}_m, \quad (4)$$

where $\hat{C} = \operatorname{argmax}_{C \subseteq \mathcal{Y}} \det(\mathbf{L}_C)$, and $\hat{C}_i = \operatorname{argmax}_{C_i \subseteq \mathcal{Y}_i} \det(\mathbf{L}_{C_i})$.

Theorem 1 tells us that the MAP inference with a strictly block diagonal kernel can be decomposed into a series of sub-inferences with its sub-kernels. In this way, the overall computation cost can be largely reduced. Noting that no exact DPP-MAP algorithms are available so far, any approximate DPP-MAP algorithms could be used in a plug-and-play way for the sub-inferences.

Almost Block Diagonal Kernel

Now we analyze the MAP inference for BwDPP with an almost block diagonal kernel as defined in (3). Let $C \subseteq \mathcal{Y}$ be the hypothesized subset to be selected from \mathbf{L} and let $C_1 \subseteq \mathcal{Y}_1, \dots, C_m \subseteq \mathcal{Y}_m$ be that from $\mathbf{L}_1, \dots, \mathbf{L}_m$ correspondingly, where $C_i = C \cap \mathcal{Y}_i$. Without loss of generality, we assume \mathbf{L}_{C_i} is invertible² for $i = 1, \dots, m$. By defining $\tilde{\mathbf{L}}_{C_i}$ recursively as $\tilde{\mathbf{L}}_{C_i} \triangleq$

$$\begin{cases} \mathbf{L}_{C_i} & i = 1, \\ \mathbf{L}_{C_i} - \mathbf{L}_{C_{i-1}, C_i}^T \tilde{\mathbf{L}}_{C_{i-1}}^{-1} \mathbf{L}_{C_{i-1}, C_i} & i = 2, \dots, m \end{cases} \quad (5)$$

one could rewrite the MAP objective function: $\det(\mathbf{L}_C)$

$$\begin{aligned} &= \det(\mathbf{L}_{C_1}) \det(\mathbf{L}_{\cup_{i=2}^m C_i} - \mathbf{L}_{C_1, \cup_{i=2}^m C_i}^T \mathbf{L}_{C_1}^{-1} \mathbf{L}_{C_1, \cup_{i=2}^m C_i}) \\ &= \det(\tilde{\mathbf{L}}_{C_1}) \det\left(\begin{bmatrix} \tilde{\mathbf{L}}_{C_2} & [\mathbf{L}_{C_2, C_3} \mathbf{0}] \\ [\mathbf{L}_{C_2, C_3} \mathbf{0}]^T & \mathbf{L}_{\cup_{i=3}^m C_i} \end{bmatrix}\right), \end{aligned} \quad (6)$$

where $\mathbf{0}$ represents zero matrix of appropriate size that fill the corresponding area with zeros. The key to the second equation above is $\mathbf{L}_{C_1, C_i} = \mathbf{0}$ for $i \geq 3$, since \mathbf{L} is an almost block diagonal kernel. Continuing this recursion,

$$\det(\mathbf{L}_C) = \dots = \prod_{i=1}^m \det(\tilde{\mathbf{L}}_{C_i}). \quad (7)$$

Hence, the MAP objective function is reduced to:

$$\operatorname{argmax}_{C \subseteq \mathcal{Y}} \det(\mathbf{L}_C) = \operatorname{argmax}_{C_1 \in \mathcal{Y}_1, \dots, C_m \in \mathcal{Y}_m} \prod_{i=1}^m \det(\tilde{\mathbf{L}}_{C_i}). \quad (8)$$

As $\tilde{\mathbf{L}}_{C_i}$ depends on C_1, \dots, C_i , we cannot optimize $\det(\tilde{\mathbf{L}}_{C_1}), \dots, \det(\tilde{\mathbf{L}}_{C_m})$ separately. Alternatively, we provide an approximate method that optimize over C_1, \dots, C_m sequentially, named the BwDPP-MAP method, which is a depth-first greedy search method in essence. The BwDPP-MAP is described in Table 1, where $\operatorname{argmax}_{C_i; C_j = \hat{C}_j, j=1, \dots, i-1}$ denotes optimizing over C_i with the value of C_j fixed as \hat{C}_j for $j = 1, \dots, i-1$, and the sub-kernel³ $\tilde{\mathbf{L}}_{\mathcal{Y}_i}$ is given similarly as $\tilde{\mathbf{L}}_{C_i}$, namely $\tilde{\mathbf{L}}_{\mathcal{Y}_i} \triangleq$

$$\begin{cases} \mathbf{L}_i & i = 1, \\ \mathbf{L}_i - \mathbf{L}_{C_{i-1}, \mathcal{Y}_i}^T \tilde{\mathbf{L}}_{C_{i-1}}^{-1} \mathbf{L}_{C_{i-1}, \mathcal{Y}_i} & i = 2, \dots, m \end{cases} \quad (9)$$

One may notice that $(\tilde{\mathbf{L}}_{\mathcal{Y}_i})_{C_i}$ is equivalent to $\tilde{\mathbf{L}}_{C_i}$.

²That simply assumes that we only consider the non-trivial subsets selected with a DPP kernel \mathbf{L} , i.e. $\det(\mathbf{L}_{C_i}) > 0$.

³Both $\mathbf{L}_{\mathcal{Y}_i}$ and $\tilde{\mathbf{L}}_{\mathcal{Y}_i}$ are called sub-kernels.

Table 1: BwDPP-MAP Algorithm

| | |
|----------------|---|
| Input: | \mathbf{L} as defined in (3); |
| Output: | Subset of items \hat{C} . |
| For: | $i = 1, \dots, m$ |
| | Compute $\tilde{\mathbf{L}}_{\mathcal{Y}_i}$ via (9); |
| | Perform sub-inference over C_i via |
| | $\hat{C}_i = \operatorname{argmax}_{C_i \in \mathcal{Y}_i; C_j = \hat{C}_j, j=1, \dots, i-1} \det((\tilde{\mathbf{L}}_{\mathcal{Y}_i})_{C_i});$ |
| Return: | $\hat{C} = \bigcup_{i=1}^m \hat{C}_i$. |

In conclusion, similar to the MAP inference with a strictly block diagonal kernel, by using BwDPP-MAP, the MAP inference for an almost block diagonal kernel can be decomposed into a series of sub-inferences for the sub-kernels as well. There are four comments for this conclusion.

First, it should be noted that the above BwDPP-MAP method is an approximate optimization method, even if each sub-inference step is conducted exactly. This is because $\tilde{\mathbf{L}}_{C_i}$ depends on C_1, \dots, C_i . We provide an empirical evaluation later, showing that through block-wise operation, the greedy search in BwDPP-MAP can achieve computation speed-up with marginal sacrifice of the accuracy.

Second, by the following Lemma 1, we show that each sub-kernel $\tilde{\mathbf{L}}_{\mathcal{Y}_i}$ is positive semi-definite, so that it is theoretically guaranteed that we can conduct each sub-inference via existing DPP-MAP algorithms, e.g. the greedy DPP-MAP algorithm (Table 2) (Gillenwater, Kulesza, and Taskar 2012). One may find the proof of Lemma 1 in the appendix.

Lemma 1 $\tilde{\mathbf{L}}_{\mathcal{Y}_i} \succeq 0$, for $i = 1, \dots, m$.

Third, in order to apply BwDPP-MAP, we need to first partition a given DPP kernel into the form of an almost block diagonal matrix as defined in (3). The partition is not unique. A trivial partition for an arbitrary DPP kernel is no partition, i.e., regarding the whole matrix as a single block. We leave the study of finding the optimal partition for further work. Here we provide a heuristic rule for partition, which is called γ -partition and performs well in our experiments.

Definition 1 (γ -partition) A γ -partition is defined by partitioning a DPP kernel \mathbf{L} into the almost block diagonal form as defined in (3) with the maximum number of blocks (i.e. the largest possible m)⁴, where for every off-diagonal matrix \mathbf{A}_i , the size of its non-zero area is only at the bottom left and does not exceed $\gamma \times \gamma$.

A heuristic way to obtain γ -partition for a kernel \mathbf{L} is to first identify a series of non-overlapping dense square sub-matrices along the main diagonal as many as possible. Next, two adjacent square sub-matrices in the main diagonal are merged if the size of the non-zero area in their corresponding off-diagonal sub-matrix exceeds $\gamma \times \gamma$.

It should be noted that a kernel could be subject to γ -partition in one or more ways with different values of γ . By taking γ -partition for a kernel with different values of γ , we can obtain a balance between computation cost and optimization accuracy. A smaller γ implies smaller m achieved.

⁴Generally speaking, a partition of a kernel of size N into m sub-kernels will approximately reduce the computational complexity m^2 times. A larger m implies larger computation reduction.

able in γ -partition, and thus smaller computation reduction. On the other hand, a smaller γ means smaller degree of interaction between adjacent sub-inferences, and thus better optimization accuracy.

Fourth, an empirical illustration of BwDPP-MAP is given in Fig. 2, where the greedy MAP algorithm (Table 2) (Gillenwater, Kulesza, and Taskar 2012) is used for the sub-inferences in BwDPP-MAP. The synthetic kernel size is fixed as 500. For each realization, the area of non-zero entries in the kernel is first specified by uniformly randomly choosing the size of sub-kernels from $[10, 30]$ and the size of the non-zero areas in off-diagonal sub-matrices from $\{0, 2, 4, 6\}$. Next, a vector \mathbf{B}_i is generated for each item i separately, following standard normal distribution. Finally, for all non-zero entries ($L_{ij} \neq 0$) specified in the previous step, the entry value is given by $L_{ij} = \mathbf{B}_i^T \mathbf{B}_j$. Fig. 2 (a) provides an example for such synthetic kernel.

We generate 1000 synthetic kernels as described above. For each synthetic kernel, we take γ -partition with $\gamma = 0, 2, 4, 6$, and then run BwDPP-MAP. The performance of directly applying the greedy MAP algorithm on the original unpartitioned kernel is used as baseline. The results in Fig. 2 (b) show that BwDPP-MAP runs much faster than the baseline. With the increase of γ , the runtime drops while the inference accuracy degrades within a tolerable range.

Connection between BwDPP-MAP and its Sub-inference Algorithm

Any DPP-MAP inference algorithm can be used in a plug-and-play fashion for the sub-inference procedure of BwDPP. It is natural to ask the connection between BwDPP-MAP and its corresponding DPP-MAP algorithm. The relation is given by the following result.

Theorem 2 *Let f be any DPP-MAP algorithm for BwDPP-MAP sub-inference, where f maps a positive semi-definite matrix to a subset of its indices, i.e. $f : \mathbf{L} \in \mathbb{S}_+ \mapsto Y \subseteq \mathcal{Y}$. BwDPP-MAP (table 1) is equivalent to applying the following steps successively to the almost block diagonal kernel as defined in (3):*

$$\hat{C}_1 = f(\mathbf{L}_{\mathcal{Y}_1}), \quad (10)$$

and for $i = 2, \dots, m$,

$$\hat{C}_i = f(\mathbf{L}_{\cup_{j=1}^i \mathcal{Y}_j} | \hat{C}_{1:i-1} \subseteq Y, \bar{\hat{C}}_{1:i-1} \cap Y = \emptyset). \quad (11)$$

where $\hat{C}_{1:i-1} = \cup_{j=1}^{i-1} \hat{C}_j$, $\bar{\hat{C}}_{1:i-1} = \cup_{j=1}^{i-1} (\mathcal{Y}_j / \hat{C}_j)$, and the input of f is the conditional kernel⁵.

The proof of Theorem 2 is in the appendix. Theorem 2 states that BwDPP-MAP is essentially a series of Bayesian belief updates, where in each update a conditional kernel is fed into f that contains the information of previous selection

⁵The conditional distribution (over set $\mathcal{Y} - A^{in} - A^{out}$) of the DPP defined by \mathbf{L} ,

$$\mathcal{P}_{\mathbf{L}}(Y = A^{in} \cup B | A^{in} \subseteq Y, A^{out} \cap Y = \emptyset), \quad (12)$$

is also a DPP (Kulesza and Taskar 2012), and the corresponding kernel, $(\mathbf{L} | A^{in} \subseteq Y, A^{out} \cap Y = \emptyset)$, is called the conditional kernel.

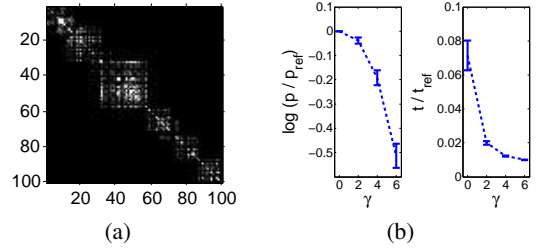


Figure 2: (a) The top-left 100×100 entries from a 500×500 synthetic kernel. (b) The log-probability ratio $\log(p/p_{\text{ref}})$ and runtime ratio t/t_{ref} , obtained from using BwDPP-MAP on the same kernel with different γ -partition, where p_{ref} and t_{ref} are the baseline performance of directly applying the greedy MAP algorithm on the original unpartitioned kernel. Results are averaged over 1000 kernels. The error bar represents 99.7% confidence level.

Table 2: Greedy DPP-MAP Algorithm

| | |
|---|----------------------------|
| Input: \mathbf{L} ; | Output: \hat{C} . |
| Initialization: Set $\hat{C} \leftarrow \emptyset, U \leftarrow \mathcal{Y}$; | |
| While U is not empty; | |
| $i^* \leftarrow \operatorname{argmax}_{i \in U} L_{ii}; \quad \hat{C} \leftarrow \hat{C} \cup \{i^*\};$ | |
| Compute $\mathbf{L}^* = \left(\left[(\mathbf{L} + \mathbf{I}_{\bar{\hat{C}}})^{-1} \right]_{\bar{\hat{C}}} \right)^{-1} - \mathbf{I}$; | |
| $\mathbf{L} \leftarrow \mathbf{L}^*; \quad U \leftarrow \{i i \notin \hat{C}, L_{ii} > 1\};$ | |
| Return: \hat{C} . | |

result. The equivalent form allows us to compare BwDPP-MAP directly with the method of applying f on the entire kernel. The latter does inference on the entire set \mathcal{Y} for one time, while the former does the inference on a sequence of smaller subsets $\mathcal{Y}_1, \dots, \mathcal{Y}_m$. Concretely, in the i -th update, a subset \mathcal{Y}_i is added to have the kernel $\mathbf{L}_{\cup_{j=1}^i \mathcal{Y}_j}$. Then the information of previous selection result is incorporated into the kernel to generate the conditional kernel. Finally, the DPP-MAP inference is performed on the conditional kernel to select \hat{C}_i from \mathcal{Y}_i .

BwDPP-based Change-Point Detection

Let $\mathbf{x}_1, \dots, \mathbf{x}_T$ be the time-series observations, where $\mathbf{x}_t \in \mathbb{R}^D$ represents the D -dimensional observation at time $t = 1, \dots, T$, and let $\mathbf{x}_{\tau:t}$ denote the segment of observations in the time interval $[\tau, t]$. We further use $\mathbf{X}_1, \mathbf{X}_2$ to represent different segments of observations at different intervals, when explicitly denoting the beginning and ending times of the intervals are not necessary. The new CPD method will build on existing metrics. A dissimilarity metric is denoted as $d : (\mathbf{X}_1, \mathbf{X}_2) \mapsto \mathbb{R}$, which measures the dissimilarity between two arbitrary segments \mathbf{X}_1 and \mathbf{X}_2 .

Quality-Diversity Decomposition of Kernel

Given a set of items $\mathcal{Y} = \{1, \dots, N\}$, the DPP kernel \mathbf{L} can be written as a Gram matrix $\mathbf{L} = \mathbf{B}^T \mathbf{B}$, where \mathbf{B}_i , the columns of \mathbf{B} , are vectors representing items in \mathcal{Y} .

A popular decomposition of the kernel is to define $\mathbf{B}_i = q_i \phi_i$, where $q_i \in \mathbb{R}^+$ measures the quality (magnitude) of item i in \mathcal{Y} , and $\phi_i \in \mathbb{R}^k, \|\phi_i\| = 1$ can be viewed as the

angle vector of diversity features so that $\phi_i^T \phi_j$ measures the similarity between items i and j . Therefore, \mathbf{L} is defined as

$$\mathbf{L} = \text{diag}(\mathbf{q}) * \mathbf{S} * \text{diag}(\mathbf{q}), \quad (13)$$

where \mathbf{q} is the quality vector consisting of q_i , and \mathbf{S} is the similarity matrix consisting of $S_{ij} = \phi_i^T \phi_j$. The quality-diversity decomposition allows us to construct \mathbf{q} and \mathbf{S} separately to address different concerns, which is utilized below to construct the kernel for CPD.

BwDppCpd

BwDppCpd is a two-step CPD method, described as follows.

Step 1: Based on a dissimilarity metric d , a preliminary set of change-point candidates is created. Consider moving a pair of adjacent windows, $\mathbf{x}_{t-w+1:t}$ and $\mathbf{x}_{t+1:t+w}$, along $t = w, \dots, T - w$, where w is the size of local windows. Then, a large d value for the adjacent windows, i.e. $d(\mathbf{x}_{t-w+1:t}, \mathbf{x}_{t+1:t+w})$, suggests that a change-point is likely to occur at time t . After we obtain the series of d values, local peaks above the mean of the d values are marked and the corresponding locations, say t_1, \dots, t_N , are selected to form the preliminary set of change-point candidates $\mathcal{Y} = \{1, \dots, N\}$.

Step 2: Treat the change-point candidates $\mathcal{Y} = \{1, \dots, N\}$ as BwDPP items, and select a subset from them to be our final estimate of the change-points.

The BwDPP kernel is built via quality-diversity decomposition. We use the similarity metric d once more to measure the quality of a candidate change-point to be a true one. Specifically, we define

$$q_i = d(\mathbf{x}_{t_i-1:t_i}, \mathbf{x}_{t_i:t_i+1}), \quad (14)$$

The higher the value q_i is, the sharper contrast around the change-point candidate i , and the better quality of i .

Next, the BwDPP similarity matrix is defined to address the fact that the true locations of change-points along the timeline tend to be diverse, since states would not change rapidly. This is done by assigning high similarity score to items being close to each other. Specifically, we define

$$S_{ij} = \exp(-(t_i - t_j)^2 / \sigma^2), \quad (15)$$

where σ is a parameter representing the position diversity level. Finally, after taking γ -partition of the kernel \mathbf{L} into the almost block diagonal form, BwDPP-MAP is used to select a set of change-points that favours both quality and diversity (Fig. 3 (b)).

Discussion

There is a rich studies of metrics for CPD problem. The choice of the dissimilarity metric $d(\mathbf{X}_1, \mathbf{X}_2)$ is flexible and could be well-tailored to the characteristics of the data. We present two examples that are used in our experiments.

- Symmetric Kullback-Leibler Divergence (SymKL):
If the two segments $\mathbf{X}_1, \mathbf{X}_2$ to be compared are assumed to follow Gaussian processes, the SymKL metric is given:

$$\text{SymKL}(\mathbf{X}_1, \mathbf{X}_2) = \text{tr}(\Sigma_1 \Sigma_2^{-1}) + \text{tr}(\Sigma_2 \Sigma_1^{-1}) - 2D + \text{tr}((\Sigma_1^{-1} + \Sigma_2^{-1})(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T), \quad (16)$$

where $\boldsymbol{\mu}$ and Σ are corresponding sample mean and covariance.

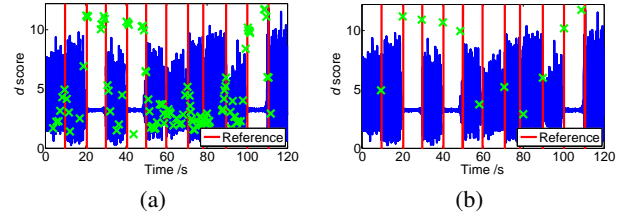


Figure 3: An BwDppCpd example from *Hasc*. (a) Change-point candidates selected in Step 1 with their d scores (green cross). (b) Final estimate of change-points in step 2 with their d scores (green cross).

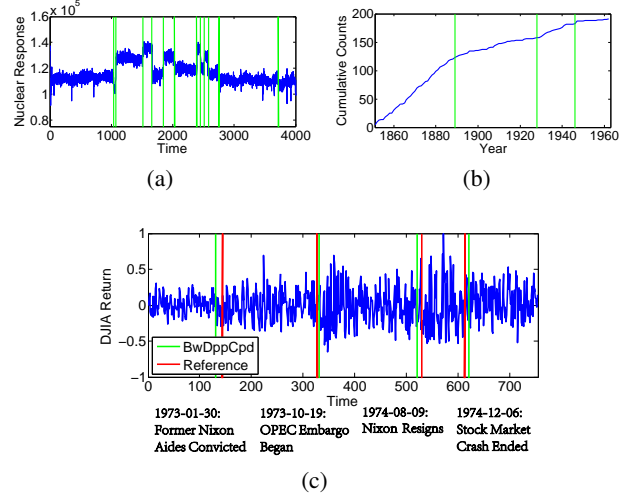


Figure 4: BwDppCpd results for *Well-Log* (a), *Coal Mine Disaster* (b), and *DJIA* (c). Green lines are detected changes.

- Generalized Likelihood Ratio (GLR):

Generally, the GLR metric is given by the likelihood ratio:

$$\text{GLR}(\mathbf{X}_1, \mathbf{X}_2) = \frac{\mathcal{L}(\mathbf{X}_1 | \lambda_1) \mathcal{L}(\mathbf{X}_2 | \lambda_2)}{\mathcal{L}(\mathbf{X}_{1,2} | \lambda_{1,2})}. \quad (17)$$

The numerator is the likelihood that the two segments follows two different models λ_1 and λ_2 respectively, while the denominator is that two segments together (denoted as $\mathbf{X}_{1,2}$) follows a single model $\lambda_{1,2}$. In practice, we plug the maximum likelihood estimates (MLE) for the parameters λ_1 , λ_2 , and $\lambda_{1,2}$. E.g. if we assume that the time-series segment $\mathbf{X} \triangleq \{x_1, \dots, x_M\}$ follows a homogeneous Poisson process, where x_i is the occurring time of the i -th event, $i = 1, \dots, M$. The log-likelihood of \mathbf{X} is

$$\mathcal{L}(\mathbf{X} | \lambda) = (M - 1) \log \lambda - (x_M - x_1) \lambda \quad (18)$$

where the MLE of λ is used, $\lambda = (M - 1) / (x_M - x_1)$.

Experiments

The BwDppCpd method are evaluated on five real-world time-series data. Firstly, three classic datasets are examined for CPD, namely *Well-Log* data, *Coal Mine Disaster* data, and *Dow Jones Industrial Average Return (DJIA)* data, where we set $\gamma = 0$ due to the small data size.

Next, we experiment with human activity detection and speech segmentation, where the data size becomes larger

| | PRC% | RCL% | F_1 |
|----------|-------|-------|--------|
| BwDppCpd | 93.05 | 87.88 | 0.9039 |
| RuLSIF | 86.36 | 83.84 | 0.8508 |

Table 3: CPD result on human activity detection data *HASC*.

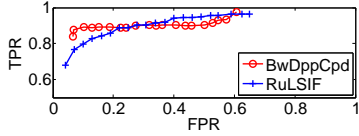


Figure 5: The ROC curve of BwDppCpd and RuLSIF.

and there is no accurate model to characterize the data, making the CPD task harder. In both experiments, the numbers of DPP items varies from hundreds to thousands, where, except BwDPP-MAP, no other algorithms can perform MAP inference within a reasonable cost of time due to the large kernel scale. We set $\gamma = 3$ for human activity detection and $\gamma = 0, 2$ for speech segmentation to provide a comparison.

As for the dissimilarity metric d , Poisson processes and GLR are used in *Coal Mine Disaster* and for other experiments, Gaussian models and SymKL are used.

Well-Log Data

Well-Log contains 4050 measurements of nuclear magnetic response taken during the drilling of a well. It is an example of varying Gaussian mean and the changes reflect the stratification of the earth’s crust (Adams and MacKay 2007). Outliers are removed prior to the experiment. As shown in Fig. 4 (a), all changes are detected by BwDppCpd.

Coal Mine Disaster Data

Coal Mine Disaster (Jarrett 1979), a standard dataset for testing CPD method, consists of 191 accidents from 1851 to 1962. The occurring rates of accidents are believed to have changed a few times and the task is to detect them. The BwDppCpd detection result, as shown in Fig. 4 (b), agrees with that in (Green 1995).

1972-75 Dow Jones Industrial Average Return

DJIA contains daily return rates of Dow Jones Industrial Average from 1972 to 1975. It is an example of varying Gaussian variance, where the changes are caused by big events that have potential macroeconomic effects. Four changes in the data are detected by BwDppCpd, which are matched well with important events (Fig. 4 (c)). Compared to (Adams and MacKay 2007), one more change is detected (the right-most), which corresponds to the date that 73-74 stock market crash ended⁶. This shows that the BwDppCpd discovers more information from the data.

Human Activity Detection

*HASC*⁷ contains human activity data collected by portable three-axis accelerometers and the task is to segment the data according to human behaviour changes. Fig. 3 (b) shows an example of *Hasc*. The performance of the best algorithm

in (Liu et al. 2013), RuLSIF, is used for comparison and the precision (PRC), recall (RCL), and F_1 measure (Kotti, Moschou, and Kotropoulos 2008) are used for evaluation:

$$\text{PRC} = \text{CFC}/\text{DET}, \quad \text{RCL} = \text{CFC}/\text{GT}, \quad (19)$$

$$F_1 = 2 \text{ PRC RCL} / (\text{PRC} + \text{RCL}), \quad (20)$$

where CFC is the number of correctly found changes, DET is the number of detected changes, and GT is the number of ground-truth changes. F_1 score could be viewed as a overall score that balances PRL and RCL. The CPD result is shown in Table 3, where the parameters are set to attain the best F_1 results for both algorithms.

The receiver operating characteristic (ROC) curve is often used to evaluate performance under different precision and recall, where true positive rate (TPR) and false positive rate (FPR) are given by $\text{TPR} = \text{RCL}$ and $\text{FPR} = 1 - \text{PRC}$. For BwDppCpd, different levels of TPR and FPR are obtained by tuning the position diversity parameter σ and for RuLSIF by tuning the threshold η (Liu et al. 2013).

As shown in Table 3 and Fig. 5, BwDppCpd outperforms RuLSIF on *HASC* when the FPR is low. RuLSIF has a better performance only when FPR exceeds 0.3, which is less useful.

Speech Segmentation

We tested two datasets for speech segmentation. The first dataset, called *Hub4m97*, is a subset (around 5 hours) from 1997 Mandarin Broadcast News Speech (HUB4-NE) released by LDC⁸. The second dataset, called *TelRecord*, consists of 216 telephone conversations, each around 2-min long, collected from real-world call centres. Acoustic features of 12-order MFCCs (mel-frequency cepstral coefficients) are extracted as the time-series data.

Speech segmentation is to segment the audio data into acoustically homogeneous segments, e.g. utterances from a single speaker or non-speech portions. The two datasets contain utterances with hesitations and a variety of changing background noises, presenting a great challenge for CPD.

The BwDppCpd method with different γ for kernel partition (denoted as Bw- γ in Table 4) is tested and two classic segmentation methods BIC (Chen and Gopalakrishnan 1998) and DISTBIC (Delacourt and Wellekens 2000) are used for comparison. As the same as in (Delacourt and Wellekens 2000), a post-processing step based on BIC values is also taken to reduce the false alarms for BwDppCpd.

The experiment results in Table 4 shows that BwDppCpd outperforms BIC and DISTBIC for both datasets. In addition, comparing the results obtained with $\gamma = 0$ and $\gamma = 2$, using $\gamma = 2$ is found to be faster but has a slightly worse performance. This agrees with our analysis of BwDPP-MAP for using different γ -partition to tradeoff speed and accuracy.

Conclusion

In this paper, we introduced BwDPPs, a class of DPPs where the kernel is almost block diagonal and thus can allow efficient block-wise MAP inference. Moreover, BwDPPs are

⁶http://en.wikipedia.org/wiki/1973-74_stock_market_crash

⁷<http://hasc.jp/hc2011/>

⁸<http://catalog.ldc.upenn.edu/LDC98S73>

| | BIC | DistBIC | Bw-0 | Bw-2 |
|------------------|--------|---------|--------|--------|
| <i>Hub4m97</i> | | | | |
| PRC% | 59.40 | 64.29 | 65.29 | 65.12 |
| RCL% | 78.24 | 74.98 | 78.49 | 78.39 |
| F_1 | 0.6753 | 0.6922 | 0.7128 | 0.7114 |
| <i>TelRecord</i> | | | | |
| PRC% | 54.05 | 61.39 | 66.54 | 66.47 |
| RCL% | 79.97 | 81.72 | 85.47 | 84.83 |
| F_1 | 0.6451 | 0.7011 | 0.7483 | 0.7454 |

Table 4: Segmentation results on *Hub4m97* and *TelRecord*.

demonstrated to be useful in change-point detection problem. The BwDPP-based change-point detection method, BwDppCpd, shows superior performance in experiments with several real-world datasets.

The almost block diagonal kernels suit the change-point detection problem well, but BwDPPs may achieve more than that. Theoretically, BwDPP-MAP could be applied to any block tridiagonal matrices without modification. It remains to be studied the theoretical issues regarding exact or approximate partition of a DPP kernel into the form of an almost block diagonal matrix (Acer, Kayaaslan, and Aykanat 2013). Other potential BwDPP applications are also worth further exploration.

Appendix: Proof of Lemma 1

PROOF Define

$$\mathbf{S}^i = \begin{cases} \mathbf{L} & i = 0 \\ \begin{bmatrix} \tilde{\mathbf{L}}_{\mathcal{Y}_{i+1}} & [\mathbf{L}_{\mathcal{Y}_{i+1}, \mathcal{Y}_{i+2}} \mathbf{0}] \\ [\mathbf{L}_{\mathcal{Y}_{i+1}, \mathcal{Y}_{i+2}} \mathbf{0}]^T & \mathbf{L}_{\cup_{j=i+2}^m \mathcal{Y}_j} \end{bmatrix} & i = 1, \dots, m-2 \\ \tilde{\mathbf{L}}_{\mathcal{Y}_{i+1}} & i = m-1 \end{cases} \quad (21)$$

For $i = 1, \dots, m-1$, \mathbf{S}^i is the Schur complement of $\tilde{\mathbf{L}}_{C_i}$ in $\mathbf{S}^{i-1}_{C_i \cup (\cup_{j=i+1}^m \mathcal{Y}_j)}$, the sub-matrix of \mathbf{S}^{i-1} . We next prove the lemma using the first principle of mathematical induction. State the predicate as:

- $P(i)$: \mathbf{S}^{i-1} and $\tilde{\mathbf{L}}_{\mathcal{Y}_i}$ are positive semi-definite (PSD).

$P(1)$ trivially holds as $\tilde{\mathbf{L}}_{\mathcal{Y}_1} = \mathbf{L}_1$ and $\mathbf{S}^0 = \mathbf{L}$ are PSD.

Assuming $P(i)$ holds. $\mathbf{S}^{i-1}_{C_i \cup (\cup_{j=i+1}^m \mathcal{Y}_j)}$ is PSD because \mathbf{S}^{i-1} is PSD. Since $\tilde{\mathbf{L}}_{C_i} \succ 0$ (footnote 2) and \mathbf{S}^i is the Schur complement of $\tilde{\mathbf{L}}_{C_i}$ in $\mathbf{S}^{i-1}_{C_i \cup (\cup_{j=i+1}^m \mathcal{Y}_j)}$, \mathbf{S}^i is PSD. Being sub-matrix of \mathbf{S}^i , $\tilde{\mathbf{L}}_{\mathcal{Y}_{i+1}}$ is also PSD. Hence, $P(i+1)$ holds.

Therefore, for $i = 1, \dots, m$, $\tilde{\mathbf{L}}_{\mathcal{Y}_i}$ is PSD. ■

Appendix: Proof of Theorem 2

For preparation, first I need to quote a result from (Kulesza and Taskar 2012): the conditional kernel

$$(\mathbf{L}|A^{in} \subseteq Y, A^{out} \cap Y = \emptyset) = ([(\mathbf{L}_{\bar{A}^{out}} + \mathbf{I}_{\bar{A}^{in}})^{-1}]_{\bar{A}^{in}})^{-1} - \mathbf{I}. \quad (22)$$

Next I need to use the following lemma:

Lemma 2 $(\mathbf{L}_{\hat{C}_{1:i}}^{-1})_{\hat{C}_i} = \tilde{\mathbf{L}}_{\hat{C}_i}^{-1}$, for $i = 1, \dots, m$, where $\tilde{\mathbf{L}}_{\hat{C}_i}$ is defined by (5).

PROOF The proof is given by mathematical induction. When $n = 1$, the result trivially holds:

$$(\mathbf{L}_{\hat{C}_1}^{-1})_{\hat{C}_1} = \mathbf{L}_{\hat{C}_1} = \tilde{\mathbf{L}}_{\hat{C}_1}^{-1}. \quad (23)$$

Assume the result holds for $n = i-1$, i.e.,

$$(\mathbf{L}_{\hat{C}_{1:i-1}}^{-1})_{\hat{C}_{i-1}} = \tilde{\mathbf{L}}_{\hat{C}_{i-1}}^{-1}. \quad (24)$$

Consider the case when $n = i$. One has

$$\begin{aligned} (\mathbf{L}_{\hat{C}_{1:i}}^{-1})_{\hat{C}_i} &= (\mathbf{L}_{\hat{C}_i} - \mathbf{L}_{\hat{C}_{1:i-1}, \hat{C}_i}^T \mathbf{L}_{\hat{C}_{1:i-1}}^{-1} \mathbf{L}_{\hat{C}_{1:i-1}, \hat{C}_i})^{-1} \\ &= (\mathbf{L}_{\hat{C}_i} - \mathbf{L}_{\hat{C}_{i-1}, \hat{C}_i}^T (\mathbf{L}_{\hat{C}_{1:i-1}}^{-1})_{\hat{C}_{i-1}} \mathbf{L}_{\hat{C}_{i-1}, \hat{C}_i})^{-1} \\ &= (\mathbf{L}_{\hat{C}_i} - \mathbf{L}_{\hat{C}_{i-1}, \hat{C}_i}^T \tilde{\mathbf{L}}_{\hat{C}_{i-1}}^{-1} \mathbf{L}_{\hat{C}_{i-1}, \hat{C}_i})^{-1} = \tilde{\mathbf{L}}_{\hat{C}_i}^{-1}. \end{aligned} \quad (25)$$

Therefore the result holds for $i = 1, \dots, m$. ■

To prove Theorem 2, it suffices to show that

$$\tilde{\mathbf{L}}_{\mathcal{Y}_i} = (\mathbf{L}_{\cup_{j=1}^i \mathcal{Y}_j} | \hat{C}_{1:i-1} \subseteq Y, \tilde{\mathcal{C}}_{1:i-1} \cap Y = \emptyset). \quad (26)$$

Using (22) one has

$$\begin{aligned} &(\mathbf{L}_{\cup_{j=1}^i \mathcal{Y}_j} | \hat{C}_{1:i-1} \subseteq Y, \tilde{\mathcal{C}}_{1:i-1} \cap Y = \emptyset) \\ &= \left(\left[(\mathbf{L}_{\hat{C}_{1:i-1} \cup \mathcal{Y}_i} + \mathbf{I}_{\mathcal{Y}_i})^{-1} \right]_{\mathcal{Y}_i} \right)^{-1} - \mathbf{I} \\ &= \mathbf{L}_{\mathcal{Y}_i} - \mathbf{L}_{\hat{C}_{1:i-1}, \mathcal{Y}_i}^T \mathbf{L}_{\hat{C}_{1:i-1}}^{-1} \mathbf{L}_{\hat{C}_{1:i-1}, \mathcal{Y}_i} \\ &= \mathbf{L}_{\mathcal{Y}_i} - \mathbf{L}_{\hat{C}_{i-1}, \mathcal{Y}_i}^T (\mathbf{L}_{\hat{C}_{1:i-1}}^{-1})_{\hat{C}_{i-1}} \mathbf{L}_{\hat{C}_{i-1}, \mathcal{Y}_i}. \end{aligned} \quad (27)$$

Following Lemma 2 to complete the proof

$$RHS = \mathbf{L}_{\mathcal{Y}_i} - \mathbf{L}_{\hat{C}_{i-1}, \mathcal{Y}_i}^T \tilde{\mathbf{L}}_{\hat{C}_{i-1}}^{-1} \mathbf{L}_{\hat{C}_{i-1}, \mathcal{Y}_i} = \tilde{\mathbf{L}}_{\mathcal{Y}_i}. \quad (28)$$

References

- [Acer, Kayaaslan, and Aykanat 2013] Acer, S.; Kayaaslan, E.; and Aykanat, C. 2013. A recursive bipartitioning algorithm for permuting sparse square matrices into block diagonal form with overlap. *SIAM Journal on Scientific Computing* 35(1):C99–C121.
- [Adams and MacKay 2007] Adams, R. P., and MacKay, D. J. 2007. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.
- [Basseville, Nikiforov, and others 1993] Basseville, M.; Nikiforov, I. V.; et al. 1993. *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs.
- [Buchbinder et al. 2012] Buchbinder, N.; Feldman, M.; Naor, J.; and Schwartz, R. 2012. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, 649–658. IEEE.
- [Chen and Gopalakrishnan 1998] Chen, S., and Gopalakrishnan, P. 1998. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 8. Virginia, USA.
- [Delacourt and Wellekens 2000] Delacourt, P., and Wellekens, C. J. 2000. Distbic: A speaker-based segmentation for audio data indexing. *Speech communication* 32(1):111–126.

- [Desobry, Davy, and Doncarli 2005] Desobry, F.; Davy, M.; and Doncarli, C. 2005. An online kernel change detection algorithm. *Signal Processing, IEEE Transactions on* 53(8):2961–2974.
- [Gillenwater, Kulesza, and Taskar 2012] Gillenwater, J.; Kulesza, A.; and Taskar, B. 2012. Near-optimal map inference for determinantal point processes. In *Advances in Neural Information Processing Systems*, 2735–2743.
- [Green 1995] Green, P. J. 1995. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika* 82(4):711–732.
- [Gustafsson and Gustafsson 2000] Gustafsson, F., and Gustafsson, F. 2000. *Adaptive filtering and change detection*, volume 1. Wiley New York.
- [Gustafsson 1996] Gustafsson, F. 1996. The marginalized likelihood ratio test for detecting abrupt changes. *Automatic Control, IEEE Transactions on* 41(1):66–78.
- [Idé and Tsuda 2007] Idé, T., and Tsuda, K. 2007. Change-point detection using krylov subspace learning. In *SDM*, 515–520. SIAM.
- [Jarrett 1979] Jarrett, R. 1979. A note on the intervals between coal-mining disasters. *Biometrika* 66(1):191–193.
- [Kanamori, Suzuki, and Sugiyama 2010] Kanamori, T.; Suzuki, T.; and Sugiyama, M. 2010. Theoretical analysis of density ratio estimation. *IEICE transactions on fundamentals of electronics, communications and computer sciences* 93(4):787–798.
- [Kawahara and Sugiyama 2012] Kawahara, Y., and Sugiyama, M. 2012. Sequential change-point detection based on direct density-ratio estimation. *Statistical Analysis and Data Mining* 5(2):114–127.
- [Kawahara, Yairi, and Machida 2007] Kawahara, Y.; Yairi, T.; and Machida, K. 2007. Change-point detection in time-series data based on subspace identification. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, 559–564. IEEE.
- [Ko, Lee, and Queyranne 1995] Ko, C.-W.; Lee, J.; and Queyranne, M. 1995. An exact algorithm for maximum entropy sampling. *Operations Research* 43(4):684–691.
- [Kotti, Moschou, and Kotropoulos 2008] Kotti, M.; Moschou, V.; and Kotropoulos, C. 2008. Speaker segmentation and clustering. *Signal processing* 88(5):1091–1124.
- [Kulesza and Taskar 2012] Kulesza, A., and Taskar, B. 2012. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*.
- [Lavielle and Lebarbier 2001] Lavielle, M., and Lebarbier, E. 2001. An application of mcmc methods for the multiple change-points problem. *Signal Processing* 81(1):39–53.
- [Liu et al. 2013] Liu, S.; Yamada, M.; Collier, N.; and Sugiyama, M. 2013. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks* 43:72–83.
- [Nemhauser, Wolsey, and Fisher 1978] Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14(1):265–294.